10GEC THE 10 GIGABIT ETHERNET CONSORTIUM

10GBASE-R PCS Test Suite V1.0

Technical Document

Last Updated: September 30, 2005 6:30pm

10 Gigabit Ethernet Consortium

University of New Hampshire
InterOperability Laboratory

121 Technology Drive, Suite 2
Durham, NH 03824
Phone: +1-603- 862-0205
Fax: +1-603-862-4181
http://www.iol.unh.edu/consortiums/10gec

MODIFICATION RECORD

- September 30, 2005 Version 1.0 Released
  - Minor procedural and editorial changes

- February, 2005 Version 0.4 Released
  - Available for external review

ACKNOWLEDGMENTS

The University of New Hampshire would like to acknowledge the efforts of the following individuals and groups in the development of this test suite.

Eric Ackerson	University of New Hampshire
David Estes	University of New Hampshire
Eric Lynskey	University of New Hampshire
Bob Noseworthy	University of New Hampshire

INTRODUCTION

Overview

The University of New Hampshire InterOperability Laboratory (UNH-IOL) is an institution designed to improve the interoperability of standards based products by providing an environment where a product can be tested against other implementations of a standard. This suite of tests has been developed to help implementers evaluate the functioning of their Clause 49 Physical Coding Sublayer (PCS) based products. The tests do not determine if a product fully conforms to IEEE Std. 802.3ae-2002. Rather, they provide one method to isolate problems within a Clause 49 PCS device. Successful completion of all tests contained in this suite does not guarantee that the tested device will operate with other devices. However, combined with satisfactory operation in the UNH-IOL interoperability test bed, these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function well in most environments.

Organization of Tests

The tests contained in this document are organized to simplify the identification of information related to a test and to facilitate in the actual testing process. Each test contains an identification section that describes the test and provides cross-reference information. The discussion section covers background information and specifies why the test is to be performed. Tests are grouped by similar functions and further organized by technology. Each test contains the following information:

Test Number

The Test Number associated with each test follows a simple grouping structure. Listed first is the Test Group Number followed by the test's number within the group. This allows for the addition of future tests to the appropriate groups of the test suite without requiring the renumbering of the subsequent tests.

Purpose

The purpose is a brief statement outlining what the test attempts to achieve. The test is written at the functional level.

References

The references section lists cross-references to IEEE 802.3 standards and other documentation that might be helpful in understanding and evaluating the test and results.

Resource Requirements

The requirements section specifies the hardware, and test equipment that will be needed to perform the test. The items contained in this section are special test devices or other facilities, which may not be available on all devices.

Last Modification

This specifies the date of the last modification to this test.

Discussion

The discussion covers the assumptions made in the design or implementation of the test as well as known limitations. Other items specific to the test are covered here.

Test Setup

The setup section describes the configuration of the test environment. Small changes in the configuration should be included in the test procedure.

Procedure

The procedure section of the test description contains the step-by-step instructions for carrying out the test. It provides a cookbook approach to testing, and may be interspersed with observable results.

Observable Results
The observable results section lists specific items that can be examined by the tester to verify that the DUT is operating properly. When multiple values are possible for an observable result, this section provides a short discussion on how to interpret them. The determination of a pass or fail for a certain test is often based on the successful (or unsuccessful) detection of a certain observable result.

Possible Problems
This section contains a description of known issues with the test procedure, which may affect test results in certain situations.

TABLE OF CONTENTS

MODIFICATION RECORD ..... **I**
ACKNOWLEDGMENTS ..... **II**
INTRODUCTION ..... **III**
TABLE OF CONTENTS ..... **V**
GROUP 1: SCRAMBLER/DESCRAMBLER AND BIT ORDERING **1**
TEST 49.1.1: TRANSMIT SCRAMBLER 2
TEST 49.1.2: RECEIVE DESCRAMBLER 3
TEST 49.1.3: TRANSMIT BIT ORDERING 4
TEST 49.1.4: RECEIVE BIT ORDERING 5
GROUP 2: CODING RULES **6**
TEST 49.2.1: 64B/66B TRANSMITTER BLOCK ENCODER 7
TEST 49.2.2: 64B/66B TRANSMITTER INVALID BLOCK HANDLING 9
TEST 49.2.3: 64B/66B RECEIVER BLOCK DECODING AND CONTROL CODE MAPPING 10
TEST 49.2.4: 64B/66B RECEIVER INVALID CODE HANDLING 12
TEST 49.2.5: IDLE CONTROL CODE INSERTION/DELETION 14
TEST 49.2.6: SEQUENCE ORDERED_SET DELETION 15
GROUP 3: LOCK STATE MACHINE **16**
TEST 49.3.1: IDENTIFICATION OF SYNC HEADER 17
TEST 49.3.2: 64_GOOD 18
TEST 49.3.3: 16_BAD 19
GROUP 4: BER MONITOR STATE MACHINE **20**
TEST 49.4.1: VALUE OF 125US_TIMER 21
GROUP 5: TRANSMIT STATE MACHINE **22**
TEST 49.5.1: IDENTIFICATION OF T_TYPE(C) 23
TEST 49.5.2: IDENTIFICATION OF T_TYPE(S) 25
TEST 49.5.3: IDENTIFICATION OF T_TYPE(T) 26
TEST 49.5.4: IDENTIFICATION OF T_TYPE(D) 27
TEST 49.5.5: IDENTIFICATION OF T_TYPE(E) 28
TEST 49.5.6: TX_INIT STATE 29
TEST 49.5.7: TX_C STATE 30
TEST 49.5.8: TX_D STATE 31
TEST 49.5.9: TX_T STATE 32
TEST 49.5.10: TX_E STATE 33
GROUP 6: RECEIVE STATE MACHINE **34**
TEST 49.6.1: IDENTIFICATION OF R_TYPE(C) 35
TEST 49.6.2: IDENTIFICATION OF R_TYPE(S) 37
TEST 49.6.3: IDENTIFICATION OF R_TYPE(T) 39
TEST 49.6.4: IDENTIFICATION OF R_TYPE(D) 40
TEST 49.6.5: IDENTIFICATION OF R_TYPE(E) 41
TEST 49.6.6: RX_INIT STATE 42
TEST 49.6.7: RX_C STATE 43
TEST 49.6.8: RX_D STATE 44
TEST 49.6.9: RX_T STATE 45
TEST 49.6.10: RX_E STATE 46
GROUP 7: TEST PATTERN VERIFICATION **48**
TEST 49.7.1: PSEUDO-RANDOM TEST PATTERN TRANSMISSION 49
TEST 49.7.2: PRBS31 TEST PATTERN TRANSMISSION 50
TEST 49.7.3: SQUARE WAVE TEST PATTERN TRANSMISSION 51
TEST 49.7.4: PSEUDO-RANDOM TEST PATTERN RECEPTION 52
TEST 49.7.5: PRBS31 TEST PATTERN RECEPTION 53
GROUP 8: MANAGEMENT **54**
GROUP 9: WIS, DELAY CONSTRAINTS AND OTHER **55**

GROUP 1: Scrambler/Descrambler and Bit Ordering

Scope: The following tests cover PCS operations specific to scrambling/descrambling and bit ordering.

Overview: These tests are designed to verify that the device under test properly scrambles/descrambles data, and properly orders data for serialization/deserialization.

Test 49.1.1: Transmit Scrambler

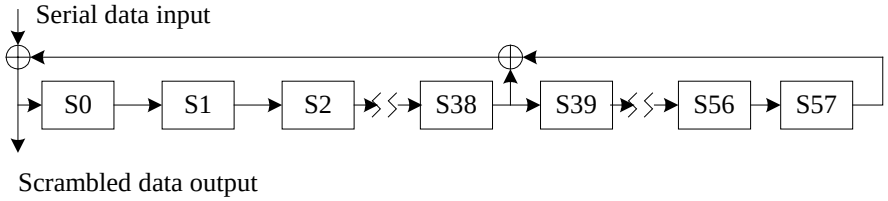
Purpose: To verify the transmit scrambler polynomial.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.6, Figure 49-8

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: February 15, 2005

Discussion: Each encoded 66-bit block is passed through a 58-tap scrambler, with the sync bits removed, in order to randomly scramble the encoded data. The sync bits, which bypass the scrambler, are then added back to each scrambled 64-bit block to form the 66-bit block that will be transmitted onto the medium. The polynomial defining the scrambler is $G(x) = 1+x^{39}+x^{58}$. There is no requirement for an initial value to the scrambler, as the receiver will automatically be able to lock to the scrambler value. After passing through the scrambler, the 66-bit blocks are cut down to 16-bit blocks and these blocks are sent to the PMA to be serialized.



Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
1. Instruct the DUT to transmit a stream of idle characters to the testing station.
2. The testing station should capture and analyze transmissions from the DUT.

Observable results:
a. The DUT should use the defined scrambler polynomial.

Possible Problems: None.

Test 49.1.2: Receive Descrambler

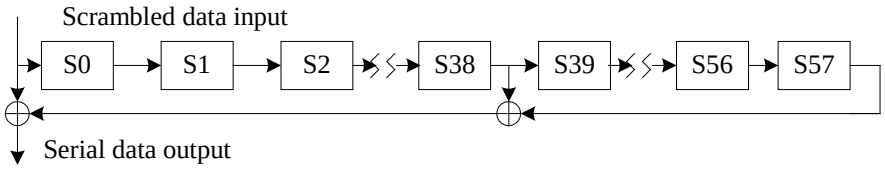
Purpose: To verify the receive descrambler polynomial.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.10, Figure 49-10

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: February 15, 2005

Discussion: On the receive side, the DUT must first deserialize the incoming bit stream into 16-bit blocks. These blocks are then analyzed such that the proper 66-bit boundary is found. Then, after stripping off the sync bits, the subsequent 64 bits are descrambled and then decoded. The descrambling polynomial used by the receiver is identical to the one used by the transmitter. Under normal conditions, the receiver will synchronize to the correct scrambler values after 64 bits have passed through the receiver. All bits following these initial 64-bits should be valid descrambled data.



Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
1. Instruct the testing station to transmit properly ordered, encoded, and scrambled idle to the DUT.
2. Observe management indicators on the DUT, checking for a valid link indication.

Observable results:
a. The DUT should be able to lock on to and recover properly scrambled data.

Possible Problems: None.

Test 49.1.3: Transmit bit ordering

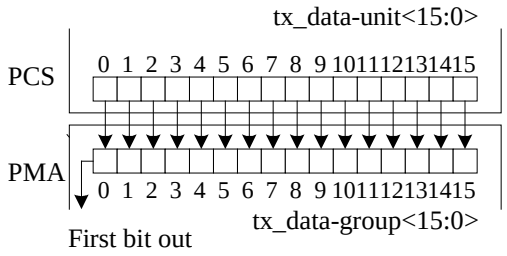
Purpose: To verify the transmit bit ordering.

References:
[1] IEEE Std 802.3ae-2002: Subclauses 49.2.7, Figures 49-2, and 49-5

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 15, 2004

Discussion: The gearbox within the PCS is needed to convert between the 66-bit blocks of the PCS and the 16-bit blocks necessary for the PMA. The gearbox takes 66-bit blocks and converts them into 16-bit tx_data-unit blocks, which are then passed to the PMA using the bit ordering shown below. Using For example, if the first sync bit of a 66-bit word maps to bit 0 of a tx_data-unit, on the following 66-bit word the sync bits will map to bits 2 and 3 of the corresponding 16-bit word.



Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
1. Instruct the DUT to transmit a stream of bits to the testing station.
2. The testing station should capture and analyze transmissions from the DUT.

Observable results:
a. The bit ordering of the DUT should match with Figures 49-2 and 49-5.

Possible Problems: None.

Test 49.1.4: Receive bit ordering

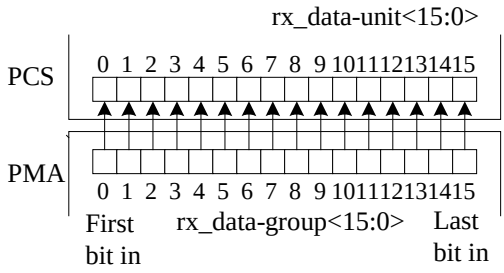
Purpose: To verify the receive bit ordering.

References:
[1] IEEE Std 802.3ae-2002: Subclauses 49.2.9, and 49.2.10, Figures 49-2, and 49-6

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 15, 2004

Discussion: On the receive side, the DUT must first deserialize the incoming bit stream into 16-bit blocks. Each serial bit is concatenated to fill up rx_data-group from bit 0 to bit 15 (the first bit received maps to bit 0). These blocks are then analyzed such that the proper 66-bit boundary is found.



Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
1. Instruct the testing station to transmit a valid bit stream to the DUT.
2. Observe management indicators on the DUT.
3. Repeat steps 1-2, instructing the testing station to transmit an otherwise valid bit stream but with bits in the wrong order (<15:0> instead of <0:15>).

Observable results:
a. The DUT should be able to lock on to and recover data with the correct bit ordering.
b. The DUT should not be able to lock on to and recover data with an incorrect bit ordering.

Possible Problems: None.

GROUP 2: Coding Rules

Scope: The following tests cover PCS operations specific to block coding/decoding.

Overview: These tests are designed to verify that the device under test properly encodes/decodes 64b/66b blocks as defined in Clause 49 of the IEEE 802.3.

Test 49.2.1: 64B/66B Transmitter Block Encoder

Purpose: To verify that the DUT can properly transmit and encode valid 66-bit blocks.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.4.6, Figure 49-7, Table 49-1

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 15, 2004

Discussion: When a device receives two consecutive transfers from the XGMII, it must encode the XGMII codes from 64-bits to 66-bits. Figure 49-7 lists all of the valid 66-bit block formats, including the sync headers, block type fields, data, and control codes. In total, there are 16 different valid data and control block combinations, as shown in Figure 49-7, for both data and control blocks. The DUT must be able to transmit all valid combinations of these blocks, and within each of the control blocks, the DUT should be able to transmit all valid combinations of the defined control codes in Table 49-1.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Valid 64B/66B Block Formats
1. Instruct the testing station to establish a valid link with the DUT.
2. Instruct the DUT to transmit valid bit streams to the testing station, containing one of the following blocks.

$D_0D_1D_2D_3/D_4D_5D_6D_7$	$O_0D_1D_2D_3/S_4D_5D_6D_7$	$T_0C_1C_2C_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/T_4C_5C_6C_7$
$C_0C_1C_2C_3/C_4C_5C_6C_7$	$O_0D_1D_2D_3/O_4D_5D_6D_7$	$D_0T_1C_2C_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/D_4T_5C_6C_7$
$C_0C_1C_2C_3/O_4D_5D_6D_7$	$S_0D_1D_2D_3/D_4D_5D_6D_7$	$D_0D_1T_2C_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/D_4D_5T_6C_7$
$C_0C_1C_2C_3/S_4D_5D_6D_7$	$O_0D_1D_2D_3/C_4C_5C_6C_7$	$D_0D_1D_2T_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/D_4D_5D_6T_7$

3. Observe all transmissions from the DUT.
4. Repeat steps 1-3, using each of the remaining valid blocks.

Part B: Control Codes
1. Instruct the testing station to establish a valid link with the DUT.
2. Instruct the DUT to transmit one of the following blocks to the testing station:

Sync	Block type	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
10	0x1E	0x2D	0x33	0x4B	0x55	0x66	0x78	0x2D	0x2D
10	0x1E	0x1E	0x1E	0x1E	0x1E	0x1E	0x1E	0x1E	0x1E

3. Observe all transmissions from the DUT.
4. Repeat steps 1-3, sending the remaining block.

Part C: O Codes
1. Instruct the testing station to establish a valid link with the DUT.
2. Instruct the DUT to transmit one of the following blocks to the testing station:

Sync	Block type	D_1	D_2	D_3	O_1	O_4	D_5	D_6	D_7
10	0x55	0x00	0x00	0x01	0x0	0x0	0x00	0x00	0x01
10	0x55	0x00	0x00	0x01	0x0	0xF	0x00	0x00	0x00
10	0x55	0x00	0x00	0x00	0xF	0xF	0x00	0x00	0x00

3. Observe all transmissions from the DUT.
4. Repeat steps 1-3, sending the remaining blocks.

Observable results:
a. The DUT should properly transmit and encode all valid 66-bit blocks.
b. The DUT should properly transmit and encode all valid control codes.
c. The DUT should properly transmit and encode all valid O codes.

Possible Problems: If an XGMII interface is not available, it may not be possible to perform all of the test cases in this test. In addition, it may take multiple attempts to see certain test vectors since there is no way to guarantee the alignment of the DUT (for example, $O_0D_1D_2D_3/S_4D_5D_6D_7$ is equivalent to $O_0D_1D_2D_3/O_4D_5D_6D_7$ followed by a $S_0D_1D_2D_3/D_4D_5D_6D_7$).

Test 49.2.2: 64B/66B Transmitter Invalid Block Handling

Purpose: To verify that the DUT does not transmit invalid blocks or control codes.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.4.6, Figure 49-7, Table 49-1

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: February 15, 2005

Discussion: When a device receives two consecutive transfers from the XGMII, it must encode the XGMII codes from 64-bits to 66-bits. Figure 49-7 lists all of the valid 66-bit block formats, including the sync headers, block type fields, data, and control codes. All block types and control codes that are not listed in these locations are considered invalid, and should not be transmitted by the DUT.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Case 1: Access to DUT through XGMII interface
1. Instruct the testing station to establish a valid link with the DUT.
2. Instruct the DUT to transmit bit streams to the testing station, containing one of the following blocks.

$C_0C_1C_2C_3/D_4D_5D_6D_7$	$C_0C_1C_2C_3/T_4C_5C_6C_7$	$D_0D_1D_2D_3/S_4D_5D_6D_7$	$D_0D_1D_2D_3/O_4D_5D_6D_7$
$T_0C_1C_2C_3/D_4D_5D_6D_7$	$T_0C_1C_2C_3/S_4D_5D_6D_7$	$T_0C_1C_2C_3/O_4D_5D_6D_7$	$S_0D_1D_2D_3/C_4C_5C_6C_7$
$S_0D_1D_2D_3/T_4C_5C_6C_7$	$S_0D_1D_2D_3/O_4D_5D_6D_7$	$O_0D_1D_2D_3/D_4D_5D_6D_7$	$O_0D_1D_2D_3/T_4C_5C_6C_7$
$T_0C_1C_2C_3/T_4C_5C_6C_7$	$S_0D_1D_2D_3/S_4D_5D_6D_7$		

3. Observe all transmissions from the DUT.
4. Repeat steps 1-3, using each of the remaining valid blocks.

Case 2: No access to DUT through XGMII interface
1. Instruct the testing station to establish a valid link with the DUT.
2. Instruct the DUT to transmit random sized frames with random contents.
3. Observe all transmissions from the DUT.

Observable results:
Case 1
a. The DUT should encode all invalid blocks as errors.

Case 2
a. The DUT should never transmit invalid blocks.

Possible Problems: If an XGMII interface is not available, it may not be possible to perform all of the test cases in this test.

Test 49.2.3: 64B/66B Receiver Block Decoding and Control Code Mapping

Purpose: To verify that the DUT can properly receive and decode valid 66-bit blocks.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.4.6, Figure 49-7, Table 49-1

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 15, 2004

Discussion: When a device has descrambled a 66-bit block, it then must decode the block from 66 to 64 bits. Figure 49-7 lists all of the valid 66-bit block formats, including the sync headers, block type fields, data, and control codes. Received blocks that do not fit in that figure, and do not include the proper information from Table 49-1, are said to be invalid blocks. All other blocks are therefore valid, and should be properly received and decoded by the DUT. In total, there are 16 different valid data and control block combinations. There are 8 control code values explicitly called out in Table 49-1, out of a possible 128, given the 7-bit length of the field. The remaining 120 control codes are all invalid if received. However, all of the listed codes, even the reserved values, must be properly received by the PCS and decoded to the correct XGMII value.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Valid 66-bit blocks and control codes
1. Instruct the testing station to establish a valid link with the DUT.
2. Instruct the testing station to transmit a valid bit pattern to the DUT, containing one of the following blocks.

$D_0D_1D_2D_3/D_4D_5D_6D_7$	$O_0D_1D_2D_3/S_4D_5D_6D_7$	$T_0C_1C_2C_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/T_4C_5C_6C_7$
$C_0C_1C_2C_3/C_4C_5C_6C_7$	$O_0D_1D_2D_3/O_4D_5D_6D_7$	$D_0T_1C_2C_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/D_4T_5C_6C_7$
$C_0C_1C_2C_3/O_4D_5D_6D_7$	$S_0D_1D_2D_3/D_4D_5D_6D_7$	$D_0D_1T_2C_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/D_4D_5T_6C_7$
$C_0C_1C_2C_3/S_4D_5D_6D_7$	$O_0D_1D_2D_3/C_4C_5C_6C_7$	$D_0D_1D_2T_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/D_4D_5D_6T_7$

3. Observe all management indications and responses from the DUT.
4. Repeat steps 1-3, using each of the remaining valid blocks.

Part B: Control Codes
1. Instruct the testing station to establish a valid link with the DUT.
2. Instruct the testing station to transmit one of the following blocks to the DUT:

Sync	Block type	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
10	0x1E	0x2D	0x33	0x4B	0x55	0x66	0x78	0x2D	0x2D
10	0x1E	0x1E	0x1E	0x1E	0x1E	0x1E	0x1E	0x1E	0x1E

3. Observe all management indications and responses from the DUT.
4. Repeat steps 1-3, sending the remaining block.

Part C: O Codes

1. Instruct the testing station to establish a valid link with the DUT.
2. Instruct the testing station to transmit one of the following blocks to the DUT:

Sync	Block type	D_1	D_2	D_3	O_1	O_4	D_5	D_6	D_7
10	0x55	0x00	0x00	0x01	0x0	0x0	0x00	0x00	0x01
10	0x55	0x00	0x00	0x01	0x0	0xF	0x00	0x00	0x00
10	0x55	0x00	0x00	0x00	0xF	0xF	0x00	0x00	0x00

3. Observe all management indications and responses from the DUT.
4. Repeat steps 1-3, sending the remaining blocks.

Observable results:

a. The DUT should properly receive and decode all valid 66-bit blocks.
b. The DUT should properly receive and decode all valid control codes.
c. The DUT should properly receive and decode all valid O codes.

Possible Problems: If access to an interface above the PCS is not available, it may not be possible to verify that the received blocks properly decoded.

Test 49.2.4: 64B/66B Receiver Invalid Code Handling

Purpose: To verify that the DUT properly handles the reception of invalid codes and blocks.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.4.6, Figure 49-7, Table 49-1

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 15, 2004

Discussion: When a device has descrambled a 66-bit block, it then must decode the block from 66 to 64 bits. Figure 49-7 lists all of the valid 66-bit block formats, including the sync headers, block type fields, data, and control codes. Received blocks that do not fit in that figure, and do not include the proper information from Table 49-1, are said to be invalid blocks. There are five conditions that indicate an invalid block exists: the sync header is invalid, the block type field contains a reserved value, an invalid control character is found, an invalid O code is found, there is no corresponding way to decode the block to the XGMII format.

Since there are only two valid possibilities for sync headers, '01' or '10', the other two combinations are invalid. Any 66-bit block that is received with a '00' or '11' sync header is therefore invalid. There are fifteen valid block type fields, each maintaining a 4-bit Hamming distance within the 8-bit field length. Each of the remaining 241 reserved block type fields are to be treated as invalid values. There are 8 control code values explicitly called out in Table 49-1, out of a possible 128, given the 7-bit length of the field. The remaining 120 control codes are all invalid if received. There are two valid O codes listed in the table, leaving 14 possible invalid O codes. Finally, there are 16 valid XGMII data and control block formats, and all others are to be received as invalid.

All invalid blocks should be completely replaced by the receiver with error control codes. These error codes are forwarded across the XGMII, and the RS will ensure that all frames received with errors are given a CRC error.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A – The sync field has a value of 00 or 11.
1. Instruct the testing station to establish a valid link with the DUT.
2. Instruct the testing station to transmit a frame to the DUT with one block of the frame using a sync header of '00' instead of '01'.
3. Observe all management indications and responses from the DUT.
4. Repeat steps 1-3, using a sync header of '11' in step 2.

Part B – The block type field contains a reserved value.
1. Instruct the testing station to establish a valid link with the DUT.
2. Instruct the testing station to transmit otherwise valid idle, but replacing the block type field of 0x1E with 0x00.
3. Observe all management indications and responses from the DUT.
4. Repeat steps 1-3, substituting the 0x00 with other reserved block type field values.

Part C – Any control character contains a value not in Table 49-1
1. Instruct the testing station to establish a valid link with the DUT.
2. Instruct the testing station to transmit otherwise valid idle, but replacing the control code fields with 0x01.
3. Observe all management indications and responses from the DUT.
4. Repeat steps 1-3, substituting the 0x01 with other control code values not found in Table 49-1.

Part D – Any O code contains a value not in Table 49-1

1. Instruct the testing station to establish a valid link with the DUT.
2. Instruct the testing station to transmit continuous sequence ordered_sets to the DUT, using a 10GBASE-R O code of 0x0, and indicating local fault.
3. Observe all management indications and responses from the DUT.
4. Repeat steps 1-3, using a 10GBASE-R O Codes of 0x1 – 0xE.

Observable results:

a. The DUT should replace the received blocks with EBLOCK_R<71:0>, and should not receive the frame.
b. The DUT should replace the received blocks with EBLOCK_R<71:0>.
c. The DUT should replace the received blocks with EBLOCK_R<71:0>.
d. The DUT should replace the received blocks with EBLOCK_R<71:0>, and should not indicate reception of local fault when receiving O codes that do not have a value of 0x0.

Possible Problems: If access to an interface above the PCS is not available, it may not be possible to verify that the received blocks are replaced with EBLOCK_R. If the RS does not force frames received with error codes to have CRC errors, then additional indications may be needed.

Test 49.2.5: Idle Control Code Insertion/Deletion

Purpose: To verify that the DUT properly inserts and delete idle

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.4.7

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: February 15, 2005

Discussion: Idle control characters (/I/) are transmitted by the PCS when idle control characters are received from the XGMII. Idle characters may be added or deleted by the PCS to adapt between clock rates. /I/ insertion and deletion shall occur in groups of 4. /I/s may be added following idle or ordered sets. They shall not be added while data is being received. When deleting /I/s, the first four characters after a /T/ shall not be deleted.

As of February 15, 2005, this test is still under development.

Test 49.2.6: Sequence Ordered_set deletion

Purpose: To verify that the DUT properly deletes ordered_sets.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.4.10

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 15, 2004

Discussion: Sequence ordered_sets may be deleted by the PCS to adapt between clock rates. Such deletion shall only occur when two consecutive sequence ordered sets have been received, and only one of the two ordered sets may be deleted.

As of February 15, 2005, this test is still under development.

GROUP 3: Lock State Machine

Scope: The following tests cover PCS operations specific to the Lock state machine (Figure 49-12)

Overview: These tests are designed to verify that the device under test properly compiles to and implements the Lock state diagram as defined in Clause 49 of the IEEE 802.3 standard.

Test 49.3.1: Identification of sync header

Purpose: To verify that the DUT properly identifies valid and invalid sync headers.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-12

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 18, 2004

Discussion: Figure 49-12 defines the Lock state machine for the PCS. This state machine is used to determine the location of the sync headers, and therefore the 66-bit boundaries, within the incoming serial bit stream at the receiver of the DUT. The DUT tests various points of the bit stream and checks for two consecutive bits to be either a '01' or a '10', which signifies a valid sync header. When 64 consecutive valid sync headers are found, without any invalid sync headers, the DUT has high confidence that it has found the correct 66-bit boundary and then achieves block_lock, and can begin to decode and receive data from its link partner. When block_lock has been achieved, the DUT sets bit 3.33.15.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Valid sync headers
1. Bring the DUT to a state with the testing station such that no link is established.
2. Instruct the testing station to send a repeating pattern to the DUT containing a sync header of '10'.
3. Observe all management indications and responses from the DUT.
4. Repeat steps 1-3, using a sync header of '01'.

Part B: Invalid sync headers
1. Bring the DUT to a state with the testing station such that no link is established.
2. Instruct the testing station to send a repeating pattern to the DUT containing a sync header of '00'.
3. Observe all management indications and responses from the DUT.
4. Repeat steps 1-3, using a sync header of '11'.

Observable results:
a. The DUT should achieve block_lock while receiving an incoming bit stream with sync headers of '01' or '10'.
b. The DUT should not achieve block_lock while receiving an incoming bit stream with sync headers of '00' or '11'.

Possible Problems: If access to bit 3.33.15 is not available, this test cannot be performed.

Test 49.3.2: 64_GOOD

Purpose: To verify that the DUT needs to see 64 consecutive valid sync headers before achieving block_lock.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-12

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 16, 2004

Discussion: Figure 49-12 defines the Lock state machine for the PCS. This state machine is used to determine the location of the sync headers, and therefore the 66-bit boundaries, within the incoming serial bit stream at the receiver of the DUT. The DUT tests various points of the bit stream and checks for two consecutive bits to be either a '01' or a '10', which signifies a valid sync header. When 64 consecutive valid sync headers are found, without any invalid sync headers, the DUT has high confidence that it has found the correct 66-bit boundary and then achieves block_lock, and can begin to decode and receive data from its link partner. When block_lock has been achieved, the DUT sets bit 3.33.15.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
1. Bring the DUT to a state with the testing station such that no link is established.
2. Instruct the testing station to send a repeating idle pattern to the DUT containing 63 valid sync headers, followed by 1 invalid sync header.
3. Observe all management indications and responses from the DUT.
4. Repeat steps 1-3, increasing the number of invalid sync headers in step 2 until a few invalid sync headers have been transmitted.
5. Repeat steps 1-4, increasing the number of valid consecutive sync headers in step 2 until the DUT has been observed to achieve block_lock, and set this value to *sh_valid_cnt*.

Observable results:
a. The DUT should achieve block after receiving a continuous looping pattern of 64 or more valid sync headers, followed by a single invalid sync header.

Possible Problems: If access to bit 3.33.15 is not available, this test cannot be performed.

Test 49.3.3: 16_BAD

Purpose: To verify that the DUT needs to see 16 out of 64 invalid sync headers before slipping.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-12

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 16, 2004

Discussion: Figure 49-12 defines the Lock state machine for the PCS. This state machine is used to determine the location of the sync headers, and therefore the 66-bit boundaries, within the incoming serial bit stream at the receiver of the DUT. The DUT tests various points of the bit stream and checks for two consecutive bits to be either a '01' or a '10', which signifies a valid sync header. For any given set of 64 consecutive sync headers, if 16 are found to be invalid, the DUT has high confidence that it has not found the correct 66-bit boundary and must therefore slip, and check another location.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Determine number of consecutive invalid sync headers needed to slip
1. Bring the DUT to a state with the testing station such that no link is established.
2. Instruct the testing station to send a repeating pattern to the DUT consisting of *n* consecutive valid sync headers and *m* consecutive invalid sync headers where *m* is initially set to 15 and *n* is initially set to (a multiple of *sh_valid_cnt*)-*m*+1, such that the length of the entire sequence is (a multiple of *sh_valid_cnt*)+1.
3. Observe all management indications and responses from the DUT.
4. Repeat steps 1-3, decrementing the value of *n* and incrementing the value of *m* until the DUT loses block_lock.

Observable results:
a. The DUT should lose block after receiving 16 invalid sync headers out of a group of 64 sync headers.

Possible Problems: If access to bit 3.33.15 is not available, this test cannot be performed.

GROUP 4: BER Monitor State Machine

Scope: The following tests cover PCS operations specific to the BER Monitor state machine (Figure 49-12)

Overview: These tests are designed to verify that the device under test properly compiles to and implements the BER Monitor state diagram as defined in Clause 49 of the IEEE 802.3 standard.

Test 49.4.1: Value of 125us_timer

Purpose: To verify that the value of the 125us_timer is between 93.75μs and 126.25μs.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13.2.5, Figure 49-13

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 18, 2004

Discussion: The 125us_timer described in Figure 49-13 must take on a value of 125μs +1%, – 25%. The timer is used to set the hi_ber flag if more than 16 invalid sync headers are found within the window of 125μs.

As of February 15, 2005 this test is still under development.

GROUP 5: Transmit State Machine

Scope: The following tests cover PCS operations specific to the Transmit state machine (Figure 49-12)

Overview: These tests are designed to verify that the device under test properly compiles to and implements the Transmit state diagram as defined in Clause 49 of the IEEE 802.3 standard.

Test 49.5.1: Identification of T_TYPE(C)

Purpose: To verify that the DUT properly identifies the reception of T_TYPE(C) vectors.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-14

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 21, 2004

Discussion: One of the functions within the PCS is the T_BLOCK_TYPE function, which classifies each transmitted 72-bit tx_raw vector as belonging to one of five possible types, depending on its contents. For a vector to be classified as C it must contain eight valid control characters other than /O/, /S/, /T/, and /E/; one valid ordered_set and four valid control characters other than /O/, /S/, and /T/; or two valid ordered sets.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Eight control characters
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_C state.
2. Provide the DUT with the following XGMII bit streams to transmit:

C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
0x07	0x07	0x07	0x07	0x07	0x07	0x07	0x07
0x1C	0x3C	0x7C	0xBC	0xDC	0xF7	0x1C	0x1C

3. Monitor all management indications and transmissions of the DUT.
4. Repeat steps 1 – 3, transmitting the other test vector.

Part B: Single ordered set and 4 control characters
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_C state.
2. Provide the DUT with the following XGMII bit streams to transmit:

C_0	C_1	C_2	C_3	O_4	D_5	D_6	D_7
0x07	0x07	0x07	0x07	0x9C	0x00	0x00	0x01
0x07	0x07	0x07	0x07	0x9C	0x00	0x00	0x02
0x07	0x07	0x07	0x07	0x9C	0x12	0x34	0x56
0x1C	0x3C	0x7C	0xBC	0x9C	0x00	0x00	0x01
0x07	0x07	0x07	0x07	0x5C	0x00	0x00	0x01

O_0	D_1	D_2	D_3	C_4	C_5	C_6	C_7
0x9C	0x00	0x00	0x01	0x07	0x07	0x07	0x07
0x9C	0x00	0x00	0x02	0x07	0x07	0x07	0x07
0x9C	0x12	0x34	0x56	0x07	0x07	0x07	0x07
0x9C	0x00	0x00	0x01	0x1C	0x3C	0x7C	0xBC
0x5C	0x00	0x00	0x01	0x07	0x07	0x07	0x07

3. Monitor all management indications and transmissions of the DUT.
4. Repeat steps 1 – 3, transmitting the other test vectors.

Part C: Two ordered sets

1. Instruct the DUT to provide valid idle to the DUT so that a link is established.
2. Provide the DUT with the following XGMII bit streams to transmit:

O_0	D_1	D_2	D_3	O_4	D_5	D_6	D_7
0x9C	0x00	0x00	0x01	0x9C	0x00	0x00	0x01
0x9C	0x00	0x00	0x02	0x9C	0x00	0x00	0x02
0x9C	0x00	0x00	0x01	0x9C	0x00	0x00	0x02
0x9C	0x00	0x00	0x02	0x9C	0x00	0x00	0x01
0x9C	0x00	0x00	0x01	0x9C	0x12	0x34	0x56
0x9C	0x12	0x34	0x56	0x9C	0x00	0x00	0x01
0x9C	0x00	0x00	0x02	0x9C	0x12	0x34	0x56
0x9C	0x12	0x34	0x56	0x9C	0x00	0x00	0x02
0x5C	0x00	0x00	0x00	0x9C	0x00	0x00	0x01
0x9C	0x00	0x00	0x01	0x5C	0x00	0x00	0x00
0x5C	0x00	0x00	0x00	0x9C	0x00	0x00	0x02
0x9C	0x00	0x00	0x02	0x5C	0x00	0x00	0x00
0x9C	0x12	0x34	0x56	0x5C	0x00	0x00	0x00
0x5C	0x00	0x00	0x00	0x9C	0x12	0x34	0x56
0x5C	0x00	0x00	0x00	0x5C	0x00	0x00	0x00

3. Monitor all management indications and transmissions of the DUT.
4. Repeat steps 1 – 3, transmitting the other test vectors.

Observable results:

a. The DUT should properly encode all 72-bit vectors.
b. The DUT should properly encode all 72-bit vectors.
c. The DUT should properly encode all 72-bit vectors.

Possible Problems: If no mechanism is available to transmit data across the XGMII interface, it may not be possible to perform this test. Other methods, including the use of loopbacks may be used to complete parts of the test.

Test 49.5.2: Identification of T_TYPE(S)

Purpose: To verify that the DUT properly identifies the reception of T_TYPE(S) vectors.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-14

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 21, 2004

Discussion: One of the functions within the PCS is the T_BLOCK_TYPE function, which classifies each transmitted 72-bit tx_raw vector as belonging to one of five possible types, depending on its contents. For a vector to be classified as S it must contain an /S/ in the first or fifth character, and any characters before the S character are valid control characters other than /O/, /S/, and /T/, or form a valid ordered_set, and all characters following the /S/ are data characters.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: S in first character
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_C state.
2. Provide the DUT with the following XGMII bit streams to transmit:

S_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0xFB	0x55	0x55	0x55	0x55	0x55	0x55	0xD5

3. Monitor all management indications and transmissions of the DUT.
4. Repeat steps 1 – 3, transmitting the other test vector.

Part B: S in fifth character
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_C state.
2. Provide the DUT with the following XGMII bit streams to transmit:

C_0	C_1	C_2	C_3	S_4	D_5	D_6	D_7
0x07	0x07	0x07	0x07	0xFB	0x55	0x55	0x55
0x1C	0x3C	0x7C	0xBC	0xFB	0x55	0x55	0x55

O_0	D_1	D_2	D_3	S_4	D_5	D_6	D_7
0x9C	0x00	0x00	0x01	0xFB	0x55	0x55	0x55
0x9C	0x00	0x00	0x02	0xFB	0x55	0x55	0x55
0x9C	0x12	0x34	0x56	0xFB	0x55	0x55	0x55
0x5C	0x00	0x00	0x00	0xFB	0x55	0x55	0x55

3. Monitor all management indications and transmissions of the DUT.
4. Repeat steps 1 – 3, transmitting the other test vectors.

Observable results:
a. The DUT should properly encode all 72-bit vectors.
b. The DUT should properly encode all 72-bit vectors.

Possible Problems: If no mechanism is available to transmit data across the XGMII interface, it may not be possible to perform this test. Other methods, including the use of loopbacks may be used to complete parts of the test.

Test 49.5.3: Identification of T_TYPE(T)

Purpose: To verify that the DUT properly identifies the reception of T_TYPE(T) vectors.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-14

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 21, 2004

Discussion: One of the functions within the PCS is the T_BLOCK_TYPE function, which classifies each transmitted 72-bit tx_raw vector as belonging to one of five possible types, depending on its contents. For a vector to be classified as T it must contain a /T/ in one of the eight characters. All characters before the /T/ must be data characters, and all characters following the /T/ are valid control characters other than /O/, /S/, and /T/.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
1. Provide a bit stream across the XGMII such that the DUT is placed in the RX_D state.
2. Instruct the DUT to transmit the first of the following 72-bit vectors to the testing station, and using 0x07 for all control characters:

$T_0C_1C_2C_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/T_4C_5C_6C_7$
$D_0T_1C_2C_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/D_4T_5C_6C_7$
$D_0D_1T_2C_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/D_4D_5T_6C_7$
$D_0D_1D_2T_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/D_4D_5D_6T_7$

3. Monitor all management indications and transmissions of the DUT.
4. Repeat steps 1 – 3, transmitting the other test vectors.
5. Repeat steps 1 – 4, replacing the 0x07 with other valid control characters.

Observable results:
a. The DUT should properly encode all 72-bit vectors.

Possible Problems: If no mechanism is available to transmit data across the XGMII interface, it may not be possible to perform this test. Other methods, including the use of loopbacks may be used to complete parts of the test.

Test 49.5.4: Identification of T_TYPE(D)

Purpose: To verify that the DUT properly identifies the reception of T_TYPE(D) vectors.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-14

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 21, 2004

Discussion: One of the functions within the PCS is the T_BLOCK_TYPE function, which classifies each transmitted 72-bit tx_raw vector as belonging to one of five possible types, depending on its contents. For a vector to be classified as D it must contain eight data characters.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Transmit all valid data codes
1. Provide a bit stream across the XGMII such that the DUT is placed in the RX_D state.
2. Instruct the DUT to transmit all valid data blocks to the testing station, encompassed within valid frames.
3. Monitor all management indications and transmissions of the DUT.

Observable results:
a. The DUT should properly encode all 72-bit vectors.

Possible Problems: If no mechanism is available to transmit data across the XGMII interface, it may not be possible to perform this test. Other methods, including the use of loopbacks may be used to complete parts of the test.

Test 49.5.5: Identification of T_TYPE(E)

Purpose: To verify that the DUT properly identifies the reception of T_TYPE(E) vectors.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-14

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 21, 2004

Discussion: One of the functions within the PCS is the T_BLOCK_TYPE function, which classifies each transmitted 72-bit tx_raw vector as belonging to one of five possible types, depending on its contents. For a vector to be classified as E it must not meet the criteria for any other value (C, S, T, or D). In any block that contains errors, the entire block is replaced with 8 XGMII error control characters.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: C Block containing E
1. Provide a bit stream across the XGMII such that the DUT is placed in the RX_C state.
2. Provide the DUT with the following XGMII bit streams to transmit:

C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
0x07	0x07	0x07	0x07	0x07	0x07	0x07	0xFE
0x07	0x07	0x07	0x07	0x07	0x07	0xFE	0x07
0x07	0x07	0x07	0x07	0x07	0xFE	0x07	0x07
0x07	0x07	0x07	0x07	0xFE	0x07	0x07	0x07
0x07	0x07	0x07	0xFE	0x07	0x07	0x07	0x07
0x07	0x07	0xFE	0x07	0x07	0x07	0x07	0x07
0x07	0xFE	0x07	0x07	0x07	0x07	0x07	0x07
0xFE	0x07	0x07	0x07	0x07	0x07	0x07	0x07

3. Monitor all management indications and transmissions of the DUT.
4. Repeat steps 1 – 3, transmitting the other test vectors.

Observable results:
a. The DUT should properly encode all 72-bit vectors as T_BLOCK_TYPE(E).

Possible Problems: If no mechanism is available to transmit data across the XGMII interface, it may not be possible to perform this test. Other methods, including the use of loopbacks may be used to complete parts of the test.

Test 49.5.6: TX_INIT state

Purpose: To verify that the DUT properly behaves while in the TX_INIT state.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-14

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: February 15, 2005

Discussion: When the DUT has been reset, it returns to the TX_INIT state, and begins the continuous transmission of local fault blocks. It will remain in this state until the reset condition has passed. Under normal conditions, the DUT will not transmit local fault across the link, but a small number of these blocks will be transmitted when the DUT is in the TX_INIT state.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Transmission of local fault
1. Through MDIO access or other means, force the DUT to reset itself.
2. Monitor all management indications and transmissions of the DUT.

Part B: Transmission of words following reset
1. Through MDIO access or other means, force the DUT to reset itself.
2. While in the TX_INIT state, force the DUT to transmit an S.
3. Monitor all management indications and transmissions of the DUT.
4. Repeat steps 1 – 3, forcing the DUT to transmit the following codes: C, E, D, T.

Observable results:
a. When being reset, the DUT should continuously transmit local fault blocks.
b. The DUT should transmit the appropriate words when exiting the TX_INIT state.

Possible Problems: If no mechanism is available to transmit data across the XGMII interface, it may not be possible to perform this test. Other methods, including the use of loopbacks may be used to complete parts of the test. If access to MDIO or other means to reset the device are not available, this test cannot be performed.

Test 49.5.8: TX_D state

Purpose: To verify that the DUT properly behaves while in the TX_D state.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-14

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 21, 2004

Discussion: When valid C blocks are being observed, within the TX_C state, the DUT will enter the TX_D state in Figure 49-14 upon the transmission of an S block. The DUT will remain in the TX_D state as long as valid D blocks are being received. The DUT will transition to the TX_T state upon transmission of a T block. The DUT will transition to the TX_E state upon transmission of an E, C, or S block.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Transmission of D
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_D state.
2. Instruct the testing station to transmit a valid D block to the DUT.
3. Monitor all management indications and transmissions of the DUT.

Part B: Transmission of T
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_D state.
2. Instruct the DUT to transmit a valid T block to the testing station.
3. Monitor all management indications and transmissions of the DUT.

Part C: Reception of E or C or S
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_D state.
2. Instruct the DUT to transmit a valid E block to the test station.
3. Monitor all management indications and transmissions of the DUT.
4. Repeat steps 1 – 3, transmitting a valid C block in step 2, then again with an S block.

Observable results:
a. The DUT should properly encode all 72-bit vectors and remain in the TX_D state.
b. The DUT should properly encode all 72-bit vectors and transition to the TX_T state.
c. The DUT should properly encode all 72-bit vectors and transition to the TX_E state.

Possible Problems: If no mechanism is available to transmit data across the XGMII interface, it may not be possible to perform this test. Other methods, including the use of loopbacks may be used to complete parts of the test.

Test 49.5.9: TX_T state

Purpose: To verify that the DUT properly behaves while in the TX_T state.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-14

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 21, 2004

Discussion: When valid D blocks are being observed at the 66-bit level, the DUT will enter the TX_T state in Figure 49-14 upon the reception of a T block. Once the TX_T state is entered, the DUT will transition to either the TX_C, TX_D, or TX_E state, depending on the next code.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Transmission of C
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_T state.
2. Instruct the DUT to transmit a valid C block to the testing station.
3. Monitor all management indications and transmissions of the DUT.

Part B: Transmission of S
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_T state.
2. Instruct the DUT to transmit a valid S block to the testing station.
3. Monitor all management indications and transmissions of the DUT.

Part C: Transmission of E, D, or T
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_T state.
2. Instruct the DUT to transmit a valid E block to the testing station.
3. Monitor all management indications and transmissions of the DUT.
4. Repeat steps 1 – 3, transmitting a valid D block in step 2, then again with a T block.

Observable results:
a. The DUT should properly encode all 72-bit vectors, and transition to the TX_C state.
b. The DUT should properly encode all 72-bit vectors, and transition to the TX_D state.
c. The DUT should properly encode all 72-bit vectors, and transition to the TX_E state.

Possible Problems: If no mechanism is available to transmit data across the XGMII interface, it may not be possible to perform this test. Other methods, including the use of loopbacks may be used to complete parts of the test.

Test 49.5.10: TX_E state

Purpose: To verify that the DUT properly behaves while in the TX_E state.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-14

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 21, 2004

Discussion: When the transmitter of the DUT encounters some sort of error, or unexpected code, it will transition from any other state, to the TX_E state. While in this state, regardless of the 66-bit words coming into the transmitter, error blocks will be encoded and passed to the PMA. If errors are transmitted in the middle of a frame, the DUT may return to the TX_D state when it transmits a valid D block, or transition to the TX_T state at the end of the frame. If valid C blocks are transmitted, the DUT will transition to the TX_C state. The DUT will remain in the TX_E state if errors continue to be transmitted or if an S block is encountered.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Transmission of D
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_E state.
2. Instruct the DUT to transmit a valid D block to the testing station.
3. Monitor all management indications and transmissions of the DUT.

Part B: Transmission of C
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_E state.
2. Instruct the DUT to transmit a valid C block to the testing station.
3. Monitor all management indications and transmissions of the DUT.

Part C: Transmission of T
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_E state.
2. Instruct the DUT to transmit a valid T block to the testing station.
3. Monitor all management indications and transmissions of the DUT.

Part D: Transmission of E or S
1. Provide a bit stream across the XGMII such that the DUT is placed in the TX_E state.
2. Instruct the DUT to transmit a valid E block to the testing station.
3. Monitor all management indications and transmissions of the DUT.
4. Repeat steps 1 – 3 with an S block in step 2.

Observable results:
a. The DUT should properly encode all 72-bit vectors and transition to the TX_D state.
b. The DUT should properly encode all 72-bit vectors and transition to the TX_C state.
c. The DUT should properly encode all 72-bit vectors and transition to the TX_T state.
d. The DUT should properly encode all 72-bit vectors and remain in the TX_E state.

Possible Problems: If no mechanism is available to transmit data across the XGMII interface, it may not be possible to perform this test. Other methods, including the use of loopbacks may be used to complete parts of the test.

GROUP 6: Receive State Machine

Scope: The following tests cover PCS operations specific to the Receive state machine (Figure 49-12)

Overview: These tests are designed to verify that the device under test properly compiles to and implements the Receive state diagram as defined in Clause 49 of the IEEE 802.3 standard.

Test 49.6.1: Identification of R_TYPE(C)

Purpose: To verify that the DUT properly identifies the reception of R_TYPE(C) vectors.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-15

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: One of the functions within the PCS is the R_BLOCK_TYPE function, which classifies each received 66-bit rx_coded vector as belonging to one of five possible types, depending on its contents. For a vector to be classified as C it must contain a sync header of 10 and meet one of these conditions: block type field of 0x1E and eight valid control characters other than /E/; block type field of 0x2D or 0x4B, a valid O code, and four valid control characters; a block type field of 0x55 and two valid O codes.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Eight control characters
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_C state.
2. Instruct the testing station to transmit the first of the following 66-bit vectors to the DUT:

C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
0x07	0x07	0x07	0x07	0x07	0x07	0x07	0x07
0x1C	0x3C	0x7C	0xBC	0xDC	0xF7	0x1C	0x1C

3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
4. Repeat steps 1 – 3, transmitting the other test vector.

Part B: Single O code and 4 control characters
1. Instrtuct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_C state.
2. Instruct the testing station to transmit the first of the following 66-bit vectors to the DUT:

Sync	Block type	C_0	C_1	C_2	C_3	O_4	D_5	D_6	D_7
10	0x2D	0x00	0x00	0x00	0x00	0x0	0x00	0x00	0x01
10	0x2D	0x00	0x00	0x00	0x00	0x0	0x00	0x00	0x02
10	0x2D	0x00	0x00	0x00	0x00	0x0	0x00	0x00	0x00
10	0x2D	0x00	0x00	0x00	0x00	0xF	0x00	0x00	0x00
10	0x2D	0x2D	0x33	0x4B	0x55	0x0	0x00	0x00	0x01

Sync	Block type	D_1	D_2	D_3	O_0	C_4	C_5	C_6	C_7
10	0x4B	0x00	0x00	0x01	0x0	0x00	0x00	0x00	0x00
10	0x4B	0x00	0x00	0x02	0x0	0x00	0x00	0x00	0x00
10	0x4B	0x12	0x34	0x56	0x0	0x00	0x00	0x00	0x00
10	0x4B	0x00	0x00	0x00	0xF	0x00	0x00	0x00	0x00
10	0x4B	0x00	0x00	0x01	0x0	0x2D	0x33	0x4B	0x55

3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
4. Repeat steps 1 – 3, transmitting the other test vectors.

Part C: Two O codes

1. Instruct the DUT to provide valid idle to the DUT so that a link is established.
2. Instruct the testing station to transmit the first of the following 66-bit vectors to the DUT:

Sync	Block type	D_1	D_2	D_3	O_0	O_4	D_5	D_6	D_7
10	0x55	0x00	0x00	0x01	0x0	0x0	0x00	0x00	0x01
10	0x55	0x00	0x00	0x02	0x0	0x0	0x00	0x00	0x02
10	0x55	0x00	0x00	0x01	0x0	0x0	0x00	0x00	0x02
10	0x55	0x00	0x00	0x02	0x0	0x0	0x00	0x00	0x01
10	0x55	0x00	0x00	0x01	0x0	0x0	0x12	0x34	0x56
10	0x55	0x12	0x34	0x56	0x0	0x0	0x00	0x00	0x01
10	0x55	0x12	0x34	0x56	0x0	0x0	0x00	0x00	0x02
10	0x55	0x00	0x00	0x02	0x0	0x0	0x12	0x34	0x56
10	0x55	0x12	0x34	0x56	0xF	0x0	0x00	0x00	0x01
10	0x55	0x12	0x34	0x56	0xF	0x0	0x00	0x00	0x02
10	0x55	0x00	0x00	0x01	0x0	0xF	0x12	0x34	0x56
10	0x55	0x00	0x00	0x02	0x0	0xF	0x12	0x34	0x56
10	0x55	0x12	0x34	0x56	0xF	0xF	0x78	0x9A	0xBC
10	0x55	0x12	0x34	0x56	0x0	0x0	0x78	0x9A	0xBC

3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
4. Repeat steps 1 – 3, transmitting the other test vectors.

Observable results:

a. The DUT should properly decode all 66-bit vectors.
b. The DUT should properly decode all 66-bit vectors.
c. The DUT should properly decode all 66-bit vectors.

Possible Problems: If no mechanism is available to monitor on the XGMII interface, it may not be possible to perform this test.

Test 49.6.2: Identification of R_TYPE(S)

Purpose: To verify that the DUT properly identifies the reception of R_TYPE(S) vectors.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-15

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: One of the functions within the PCS is the R_BLOCK_TYPE function, which classifies each received 66-bit rx_coded vector as belonging to one of five possible types, depending on its contents. For a vector to be classified as S it must contain a sync header of 10 and meet one of these conditions: block type field of 0x33 and four valid control characters; a block type field of 0x66 and a valid O code; or a block type field of 0x78.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Block type field of 0x33
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_C state.
2. Instruct the testing station to transmit the first of the following 66-bit vectors to the DUT:

Sync	Block type	C_0	C_1	C_2	C_3	O_4	D_5	D_6	D_7
10	0x33	0x00	0x00	0x00	0x00	0x0	0x55	0x55	0x55
10	0x33	0x2D	0x33	0x4B	0x55	0x0	0x55	0x55	0x55

3. Instruct the testing station to transmit additional 66-bit vectors, forming a valid frame.
4. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
5. Repeat steps 1 – 3, transmitting the other test vector.

Part B: Block type field of 0x66
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_C state.
2. Instruct the testing station to transmit the first of the following 66-bit vectors to the DUT:

Sync	Block type	D_1	D_2	D_3	O_0	O_4	D_5	D_6	D_7
10	0x66	0x00	0x00	0x01	0x0	0x0	0x55	0x55	0x55
10	0x66	0x00	0x00	0x02	0x0	0x0	0x55	0x55	0x55
10	0x66	0x12	0x34	0x56	0x0	0x0	0x55	0x55	0x55
10	0x66	0x00	0x00	0x00	0xF	0x0	0x55	0x55	0x55

3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
4. Instruct the testing station to transmit additional 66-bit vectors, forming a valid frame.
5. Repeat steps 1 – 3, transmitting the other test vectors.

Part C: Block type field of 0x78
1. Instruct the DUT to provide valid idle to the DUT so that a link is established.
2. Instruct the testing station to transmit the first of the following 66-bit vectors to the DUT:

Sync	Block type	D_1	D_2	D_3	D_4	D_5	D_6	D_7
10	0x78	0x55	0x55	0x55	0x55	0x55	0x55	0xD5

3. Instruct the testing station to transmit additional 66-bit vectors, forming a valid frame.
4. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.

Observable results:
a. The DUT should properly decode all 66-bit vectors and receive the frame.
b. The DUT should properly decode all 66-bit vectors and receive the frame.
c. The DUT should properly decode all 66-bit vectors and receive the frame.

Possible Problems: If no mechanism is available to monitor on the XGMII interface, it may not be possible to perform this test.

Test 49.6.3: Identification of R_TYPE(T)

Purpose: To verify that the DUT properly identifies the reception of R_TYPE(T) vectors.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-15

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: One of the functions within the PCS is the R_BLOCK_TYPE function, which classifies each received 66-bit rx_coded vector as belonging to one of five possible types, depending on its contents. For a vector to be classified as T it must contain a sync header of 10 and meet one of these conditions: block type field of 0x87, 0x99, 0xAA, 0xB4, 0xCC, 0xD2, 0xE1, or 0xFF and all control characters are valid.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_D state.
2. Instruct the testing station to transmit the first of the following 66-bit vectors to the DUT, containing the appropriate block field types, and using 0x00 for all control characters:

$T_0C_1C_2C_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/T_4C_5C_6C_7$
$D_0T_1C_2C_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/D_4T_5C_6C_7$
$D_0D_1T_2C_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/D_4D_5T_6C_7$
$D_0D_1D_2T_3/C_4C_5C_6C_7$	$D_0D_1D_2D_3/D_4D_5D_6T_7$

3. Instruct the testing station to transmit additional idle 66-bit vectors, following the /T/.
4. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
5. Repeat steps 1 – 3, transmitting the other test vectors.
6. Repeat steps 1 – 5, replacing the 0x00 with other valid control characters.

Observable results:
a. The DUT should properly decode all 66-bit vectors and receive the frame.

Possible Problems: If no mechanism is available to monitor on the XGMII interface, it may not be possible to perform this test.

Test 49.6.4: Identification of R_TYPE(D)

Purpose: To verify that the DUT properly identifies the reception of R_TYPE(D) vectors.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-15

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: One of the functions within the PCS is the R_BLOCK_TYPE function, which classifies each received 66-bit rx_coded vector as belonging to one of five possible types, depending on its contents. For a vector to be classified as D it must contain a sync header of 01.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_D state.
2. Instruct the testing station to transmit all valid data blocks to the DUT, encompassed within valid frames.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.

Observable results:
a. The DUT should properly decode all 66-bit vectors and receive the frames.

Possible Problems: If no mechanism is available to monitor on the XGMII interface, it may not be possible to perform this test.

Test 49.6.5: Identification of R_TYPE(E)

Purpose: To verify that the DUT properly identifies the reception of R_TYPE(E) vectors.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-15

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: One of the functions within the PCS is the R_BLOCK_TYPE function, which classifies each received 66-bit rx_coded vector as belonging to one of five possible types, depending on its contents. For a vector to be classified as E it must not meet the criteria for any other value (C, S, T, or D). In any block that contains errors, the entire block is replaced with 8 XGMII error control characters.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: C Block containing E
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_C state.
2. Instruct the testing station to transmit the first of the following 66-bit vectors to the DUT:

Sync	Block type	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
10	0x1E	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x1E
10	0x1E	0x00	0x00	0x00	0x00	0x00	0x00	0x1E	0x00
10	0x1E	0x00	0x00	0x00	0x00	0x00	0x1E	0x00	0x00
10	0x1E	0x00	0x00	0x00	0x00	0x1E	0x00	0x00	0x00
10	0x1E	0x00	0x00	0x00	0x1E	0x00	0x00	0x00	0x00
10	0x1E	0x00	0x00	0x1E	0x00	0x00	0x00	0x00	0x00
10	0x1E	0x00	0x1E	0x00	0x00	0x00	0x00	0x00	0x00
10	0x1E	0x1E	0x00	0x00	0x00	0x00	0x00	0x00	0x00

3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
4. Repeat steps 1 – 3, transmitting the other test vector.

Observable results:
a. The DUT should properly decode all 66-bit vectors as R_BLOCK_TYPE(E).

Possible Problems: If no mechanism is available to monitor on the XGMII interface, it may not be possible to perform this test.

Test 49.6.6: RX_INIT state

Purpose: To verify that the DUT passes local fault across the XGMII when in the RX_INIT state.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-15

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: When the DUT has been reset, is in a test mode, is receiving frequent errors, or does not have block lock, it will remain within the RX_INIT state. While in this state, the DUT will be sending local fault blocks up across the XGMII to signify that the receiver is not operational.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Absence of block lock
1. Disconnect the receiver of the DUT, thus forcing the DUT to lose block_lock.
2. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
3. Repeat steps 1 and 2, but provide valid signaling to the DUT such that the DUT does not have block_lock.

Part B: HI BER
1. Instruct the testing station to transmit valid idle to the DUT, but inserting sync header errors every 128 blocks, thus allowing the DUT to keep block_lock but to also have hi_ber.
2. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.

Part C: Reset
1. Instruct the DUT to provide valid idle to the DUT so that a link is established.
2. Force a reset of the DUT through MDIO.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.

Part D: Receive test mode
1. Turn on a receive test mode on the DUT through MDIO access.
2. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
3. Repeat steps 1 and 2 for all available receive test modes.

Observable results:
a. When the DUT does not have block_lock, it must pass continuous local fault across the XGMII.
b. When the DUT has hi_ber, it must pass continuous local fault across the XGMII.
c. When the reset signal is set, the DUT must pass continuous local fault across the XGMII.
d. When in the receive test mode, the DUT must pass continuous local fault across the XGMII.

Possible Problems: If no mechanism is available to monitor on the XGMII interface, it may not be possible to perform this test. If MDIO access is not available, it may not be possible to do parts C and D.

Test 49.6.7: RX_C state

Purpose: To verify that the DUT properly behaves while in the RX_C state.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-15

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: When the receiver of the DUT has achieved synchronization, and when valid C blocks are being observed at the 66-bit level, the DUT will enter the RX_C state in Figure 49-15. The DUT will remain in this state as long as valid idle or sequence ordered_sets are being received. Upon the reception of an S block, the DUT will transition to the RX_D state. Reception of E, D, or T blocks will force the DUT to the RX_E state.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Reception of C
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_C state.
2. Instruct the testing station to transmit a valid C block to the DUT.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.

Part B: Reception of S
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_C state.
2. Instruct the testing station to transmit a valid S block to the DUT.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.

Part C: Reception of E, D, or T
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_C state.
2. Instruct the testing station to transmit a valid E block to the DUT.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
4. Repeat steps 1 – 3, transmitting valid D and T blocks to the DUT.

Observable results:
a. The DUT should properly decode all 66-bit vectors and remain in the RX_C state while receiving C blocks.
b. The DUT should properly decode all 66-bit vectors and transition to the RX_D state while receiving an S block.
c. The DUT should properly decode all 66-bit vectors and transition to the RX_E state when receiving E, D, or T blocks.

Possible Problems: If no mechanism is available to monitor on the XGMII interface, it may not be possible to perform this test.

Test 49.6.8: RX_D state

Purpose: To verify that the DUT properly behaves while in the RX_D state.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-15

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: When the receiver of the DUT has achieved synchronization, and when valid C blocks are being observed at the 66-bit level, the DUT will enter the RX_D state in Figure 49-15 upon the reception of an S block. The DUT will remain in the RX_D state as long as valid D blocks are being received. The DUT will transition to the RX_T state upon reception of a T block, provided that the block following the T is either an S or C block. For all other combinations of received blocks, the DUT will transition to the RX_E state.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Reception of D
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_D state.
2. Instruct the testing station to transmit a valid D block to the DUT.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.

Part B: Reception of T with proper next type
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_D state.
2. Instruct the testing station to transmit a valid T block to the DUT, followed by an S block.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
4. Repeat steps 1 – 3, transmitting a valid T block followed by a C block in step 2.

Part C: Reception of T with invalid next type
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_D state.
2. Instruct the testing station to transmit a valid T block to the DUT, followed by an E block.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
4. Repeat steps 1 – 3, transmitting a valid T block followed by a D block in step 2, then again with a T block followed by a T block.

Part D: Reception of E, C or S
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_D state.
2. Instruct the testing station to transmit a valid E block to the DUT.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
4. Repeat steps 1 – 3, transmitting a valid C block in step 2, then again with an S block.

Observable results:
a. The DUT should properly decode all 66-bit vectors and remain in the RX_D state.
b. The DUT should properly decode all 66-bit vectors and transition to the RX_T state.
c. The DUT should properly decode all 66-bit vectors and transition to the RX_E state.
d. The DUT should properly decode all 66-bit vectors and transition to the RX_E state.

Possible Problems: If no mechanism is available to monitor on the XGMII interface, it may not be possible to perform this test.

Test 49.6.9: RX_T state

Purpose: To verify that the DUT properly behaves while in the RX_T state.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-15

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: When the receiver of the DUT has achieved synchronization, and when valid D blocks are being observed at the 66-bit level, the DUT will enter the RX_T state in Figure 49-15 upon the reception of an S block. The DUT will remain in the RX_D state as long as valid D blocks are being received. The DUT will transition to the RX_T state upon reception of a T block, provided that the block following the T is either an S or C block. Once the RX_T state is entered, the DUT will transition to either the RX_C or RX_D state, depending on the next code.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Reception of C
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_T state.
2. Instruct the testing station to transmit a valid C block to the DUT.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.

Part B: Reception of S
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_T state.
2. Instruct the testing station to transmit a valid S block to the DUT.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.

Observable results:
a. The DUT should properly decode all 66-bit vectors, and transition to the RX_C state.
b. The DUT should properly decode all 66-bit vectors, and transition to the RX_D state.

Possible Problems: If no mechanism is available to monitor on the XGMII interface, it may not be possible to perform this test.

Test 49.6.10: RX_E state

Purpose: To verify that the DUT properly behaves while in the RX_E state.

References:
[1] IEEE Std 802.3ae-2002: Subclause 49.2.13, Figure 49-15

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: When the receiver of the DUT encounters some sort of error, it will transition, from any other state, to the RX_E state. While in this state, regardless of the 66-bit words coming into the receiver, error blocks will be decoded and passed up to the XGMII. If errors are received in the middle of a frame, the DUT may return to the RX_D state when it receives a valid D block, or transition to the RX_T state at the end of the frame. If valid C blocks are received, the DUT will transition to the RX_C state. The DUT will remain in the RX_E state if errors continue to be received or if an S block is encountered.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Reception of D
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_E state.
2. Instruct the testing station to transmit a valid D block to the DUT.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.

Part B: Reception of C
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_E state.
2. Instruct the testing station to transmit a valid C block to the DUT.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.

Part C: Reception of T with proper next type
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_E state.
2. Instruct the testing station to transmit a valid T block to the DUT, followed by an S block.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
4. Repeat steps 1 – 3 with a T block followed by a C block in step 2.

Part D: Reception of T with invalid next type
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_E state.
2. Instruct the testing station to transmit a valid T block to the DUT, followed by an E block.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
4. Repeat steps 1 – 3 following the T block in step 2 with a D block, and then again with a T block.

Part E: Reception of E or S
1. Instruct the testing station to provide a valid bit stream to the DUT such that it is placed in the RX_E state.
2. Instruct the testing station to transmit a valid E block to the DUT.
3. Monitor all management indications and responses of the DUT on the XGMII interface, or equivalent.
4. Repeat steps 1 – 3 with an S block in step 2.

Observable results:
a. The DUT should properly decode all 66-bit vectors and transition to the RX_D state.
b. The DUT should properly decode all 66-bit vectors and transition to the RX_C state.

c. The DUT should properly decode all 66-bit vectors and transition to the RX_T state.
d. The DUT should properly decode all 66-bit vectors and remain in the RX_E state.
e. The DUT should properly decode all 66-bit vectors and remain in the RX_E state.

Possible Problems: If no mechanism is available to monitor on the XGMII interface, it may not be possible to perform this test.

GROUP 7: Test Pattern Verification

Scope: The following tests cover PCS operations specific to test pattern generation and reception.

Overview: These tests are designed to verify that the device under test properly compiles to and implements the Test Patterns as defined in Clause 49 of the IEEE 802.3 standard.

Test 49.7.1: Pseudo-random test pattern transmission

Purpose: To verify that the DUT generates the correct pseudo-random test patterns.

References:
[1] IEEE Std 802.3ae-2002: Subclauses 45.2.3.15, 49.2.8, 52.9.1.1, Table 52-20

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 19, 2004

Discussion: Clause 49 defines a pseudo-random test pattern generator and checker to be implemented in the PCS. This functionality allows the tester to not only verify BER over a channel, but also to force the DUT into transmitting the appropriate signals used for testing. When the PCS offers a direct connection to the PMA, the implementation of the generator and checker is mandatory. The jitter test pattern is generated by the scrambler, using specific seeds that are regularly loaded through the MDIO registers, and using one of two possible data patterns to be scrambled. Every 128 blocks, the scrambler is loaded with one of two seeds, or their inverses, in the following pattern: Seed A, Seed A Invert, or Seed B, Seed B Invert. The values of the seeds are defined in Clause 52, and shown in Table 52-20. The data patterns fed through the scrambler are either all zeros, or the encoding of two Local Fault ordered_sets, and the data pattern will be inverted when the inverted seeds are being used. Register 3.42 controls the different test patterns that can be transmitted and checked.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Test pattern 1
1. Instruct the DUT to transmit Pseudo-Random Test Pattern 1.
2. Instruct the Testing Station to capture and decode all transmissions from the DUT.

Part A: Test pattern 2
1. Instruct the DUT to transmit Pseudo-Random Test Pattern 2.
2. Instruct the Testing Station to capture and decode all transmissions from the DUT.

Observable results:
a. The DUT should properly generate Pseudo-Random Test Pattern 1.
b. The DUT should properly generate Pseudo-Random Test Pattern 2.

Possible Problems: None.

Test 49.7.2: PRBS31 test pattern transmission

Purpose: To verify that the DUT generates the correct PRBS31 test pattern.

References:
[1] IEEE Std 802.3ae-2002: Subclauses 45.2.3.15, 49.2.8, 52.9.1.1, Table 52-20

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 19, 2004

Discussion: Clause 49 defines an optional PRBS31 test pattern mode that can be implemented within the PCS for both transmission and checking of the pattern. When selected, the generator sends 16 bits of the test pattern at a time down to the PMA, bypassing the scrambler. The pattern generator implements an inverted version of the bit stream generated with the following polynomial: $G(x) = 1 + x^{28} + x^{31}$. Register 3.42 controls the different test patterns that can be transmitted and checked.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: PRBS31 test pattern
1. Instruct the DUT to transmit PRBS31 test pattern.
2. Instruct the Testing Station to capture and decode all transmissions from the DUT.

Observable results:
a. The DUT should properly generate the PRBS31 test pattern.

Possible Problems: If the DUT does not support this pattern, this test cannot be done.

Test 49.7.3: Square wave test pattern transmission

Purpose: To verify that the DUT generates the correct square wave test pattern.

References:
[1] IEEE Std 802.3ae-2002: Subclauses 45.2.3.15, 49.2.8, 52.9.1.1, Table 52-20

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: Certain PMD tests that need to be performed require that the PCS generate a square wave pattern. Such a pattern necessarily needs to bypass the encoder and scrambler, and does not contain any sync bits. When the square wave pattern is selected, the PCS will send a repeating pattern of *n* ones followed by *n* zeros where *n* may be any number between 4 and 11, inclusive. The value of *n* is and implementation choice, and may be a programmable value in some devices. Register 3.42 controls the different test patterns that can be transmitted and checked.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: PRBS31 test pattern
1. Instruct the DUT to transmit a square wave test pattern.
2. Instruct the Testing Station to capture and decode all transmissions from the DUT.
3. Repeat steps 1 and 2, varying the run length of the pattern to any other values the DUT supports.

Observable results:
a. The DUT should properly generate the square wave test pattern for all values of *n*.

Possible Problems: None.

Test 49.7.4: Pseudo-random test pattern reception

Purpose: To verify that the DUT generates the correct pseudo-random test patterns.

References:
[1] IEEE Std 802.3ae-2002: Subclauses 45.2.3.15, 49.2.12, 52.9.1.1, Table 52-20

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: Clause 49 defines a pseudo-random test pattern generator and checker to be implemented in the PCS. This functionality allows the tester to not only verify BER over a channel, but also to force the DUT into transmitting the appropriate signals used for testing. When the PCS offers a direct connection to the PMA, the implementation of the generator and checker is mandatory. The jitter test pattern is generated by the scrambler, using specific seeds that are regularly loaded through the MDIO registers, and using one of two possible data patterns to be scrambled. Every 128 blocks, the scrambler is loaded with one of two seeds, or their inverses, in the following pattern: Seed A, Seed A Invert, or Seed B, Seed B Invert. The values of the seeds are defined in Clause 52, and shown in Table 52-20. The data patterns fed through the scrambler are either all zeros, or the encoding of two Local Fault ordered_sets, and the data pattern will be inverted when the inverted seeds are being used. Register 3.42 controls the different test patterns that can be transmitted and checked.

The pseudo-random checker utilizes the lock state machine and descrambler as it would during normal operations. However, the hi_ber state machine is disabled when the test pattern mode is enabled. When the output of the descrambler is the data pattern or its inverse, a match is detected. Since the transmitter's scrambler is loaded with a seed value every 128 blocks and the receiver's descrambler is running normally, a mismatch will be detected once every 128 blocks in the absence of errors. Therefore, the receiver will count in 128-block windows and ignore the first block within a window that contains a mismatch. All subsequent mismatches will be counted as errors.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: Test pattern 1
1. Instruct the testing station to transmit Pseudo-Random Test Pattern 1 to the DUT.
2. Observe all management and indications from the DUT.
3. Repeat steps 1 and 2, inserting periodic and known errors into the test pattern.

Part A: Test pattern 2
1. Instruct the testing station to transmit Pseudo-Random Test Pattern 2 to the DUT.
2. Observe all management and indications from the DUT.
3. Repeat steps 1 and 2, inserting periodic and known errors into the test pattern.

Observable results:
a. The DUT should lock on and receive Pseudo-Random Test Pattern 1, and all mismatches apart from the first in every 128-blocks, should be counted.
b. The DUT should lock on and receive Pseudo-Random Test Pattern 2, and all mismatches apart from the first in every 128-blocks, should be counted.

Possible Problems: None.

Test 49.7.5: PRBS31 test pattern reception

Purpose: To verify that the DUT generates the correct PRBS31 test pattern.

References:
[1] IEEE Std 802.3ae-2002: Subclauses 45.2.3.15, 49.2.12, 52.9.1.1, Table 52-20

Resource Requirements:
- A testing station capable of encoding (decoding) 64 bit octets to (from) 66-bit code groups as specified in Clause 49 and sending (receiving) these code groups using the signaling method described in Clause 52.

Last Modification: December 20, 2004

Discussion: Clause 49 defines an optional PRBS31 test pattern mode that can be implemented within the PCS for both transmission and checking of the pattern. When selected, the generator sends 16 bits of the test pattern at a time down to the PMA, bypassing the scrambler. The pattern generator implements an inverted version of the bit stream generated with the following polynomial: $G(x) = 1 + x^{28} + x^{31}$. Register 3.42 controls the different test patterns that can be transmitted and checked. When the receive channel is operating in PRBS31 test mode, the PRBS31 pattern checker checks the bits received from the PMA, bypassing the descrambler, and checks them relative to the PRBS31 test pattern. The pattern error checker is self synchronizing, and compares each bit received to the result of the PRBS31 generator based on the prior 31 bits received. When no errors occur, the pattern error signal will be zero. When an isolated bit error occurs, the PRBS31 pattern error signal will go high three times, and the test-pattern error counter will increment once for each bit time that the PRBS31 pattern error signal is high.

Test Setup: Connect the device under test (DUT) to the testing station (transmit to receive, receive to transmit) with the appropriate medium (i.e. single mode fiber, etc.).

Procedure:
Part A: PRBS31 test pattern
1. Instruct the testing station to transmit the PRBS31 test pattern to the DUT.
2. Observe all management and indications from the DUT.
3. Repeat steps 1 and 2, inserting known errors into the test pattern.

Observable results:
a. The DUT should properly receive the PRBS31 test pattern, and properly count all errors.

Possible Problems: If the DUT does not support this pattern, this test cannot be done.

GROUP 8: Management

Scope: The following tests cover PCS operations specific to the Clause 45 management entity.

Overview: These tests are designed to verify that the device under test properly compiles to and implements the mandatory management features defined in Clause 45 and Clause 49 of the IEEE 802.3 standard.

As of February 15, 2005 this group of tests is still under development.

GROUP 9: WIS, Delay Constraints and Other

Scope: The following tests cover PCS operations specific to the WIS and other operations not covered in previous groups of this test suite.

Overview: These tests are designed to verify that the device under test properly compiles to and implements the mandatory features for WIS, delay constraints, and other features not covered elsewhere in this test suite.

As of February 15, 2005 this group of tests is still under development.